Lab 10 Template – Ethan Roepke

Part 01

1. **"outfile-Good text" pdf and "out-file Bad text" pdf (upload the files in canvas submission)** (10 points)
   UPLOADED

2. **Screenshot of the hashes of the text PDFs colliding for SHA1 and differing with SHA512** (10 points)

```
cpre3310@cpre3310:~/homework/lab10/part01/sha1collider$ sha1sum out-goodLetter.pdf out-evilLetter.pdf
50e55c67a5c43770ab9f56cf649fdf03cc9c0cae  out-goodLetter.pdf
50e55c67a5c43770ab9f56cf649fdf03cc9c0cae  out-evilLetter.pdf
cpre3310@cpre3310:~/homework/lab10/part01/sha1collider$ sha512sum out-goodLetter.pdf out-evilLetter.pdf
39d5c0a05b51a8582f151d51b57f4aee2e2f2d1a022ec8e61ef3be61a47ba84466ebfaff1750fe3079c7fb20f1ee024029ce8a422d379bfd198b3276294c6aee  out-goodLetter.pdf
8b8ebdcd7a144083872d0a78962a2e64ddd877b2168c87ee9fda2b436e1311c84fd751e7e38b58348b00721c6d42e161616c909e9f64fc24809d41f91574c8ea  out-evilLetter.pdf
cpre3310@cpre3310:~/homework/lab10/part01/sha1collider$
```

3. **Explain why the SHA1 hashes are the same for different files, but the SHA512 hashes are different. Please be specific about how the near collision/collision works.** (10 points)
   SHA1 which is a 160 bit cryptographic hash function is vulnerable to collision attacks which means two different files can output the same hash. Attackers are able to modify a file slightly while keeping hash preserved.
   SHA512 which uses 512 bit output size makes it much harder for collision attacks. The complexity makes it hard to modify 2 different files and get the same hashes from the 2 fies.

4. **Image out-1.pdf and Image out-2.pdf (Upload the files in canvas submission)** (10points)
   UPLOADED

5. **Screenshot of the hashes of the image PDFs colliding for SHA1 and differing with SHA512** (10 points)

```
cpre3310@cpre3310:~/homework/lab10/part01/sha1collider$ sha1sum out-houseFire.pdf out-blackWhiteHouse.pdf
42813f375cdac26c84f747ca5f8a21298c1cd17b  out-houseFire.pdf
42813f375cdac26c84f747ca5f8a21298c1cd17b  out-blackWhiteHouse.pdf
cpre3310@cpre3310:~/homework/lab10/part01/sha1collider$ sha512sum out-houseFire.pdf out-blackWhiteHouse.pdf
3b05202f0b10749ccd2a9e875cf85644f76e26eb135add760caf85bae67de5cd449825ba617bc19e2358fb8048592c33b6c083b3c469c94a7cca193808928ee1  out-houseFire.pdf
ab5cf1996ec738ba26687528dcfb3cf2f5f17b99904a8423b3947eb1cb5ab481b670fbfcfd1896ffd24ae0406d1d4be4f35a3105d4f0fcc82489c5f6d651deae  out-blackWhiteHouse.pdf
cpre3310@cpre3310:~/homework/lab10/part01/sha1collider$
```

6. **How likely is this to be used to carry out an attack in the wild?** (10 total points)
   A hash collision attack can carry out an attack in our world if going for SHA1 and has a high likelihood of being successful since its very vulnerable to collision attacks. However for SHA512, its almost impossible due to the complexity of it. An example of an attack would if an organization was still relying on SHA1 and use for signature authentication for documents. An attacker can send a legit document and a malicious document that have the same hashes. An attacker can send the real document to get signed and then the attacker can replace it with malicious document since it carries the same hashes. The attacker can gain access to unauthorized documents that are supposed to be secured/private

Part 02

1. **Upload your code to canvas as a separate .py file** (10 points)
   Uploaded

2. **What is the winning lottery number for November 12th, 2024?** (5 points)
   Winning number: 8 – 0 – 4 – 1 – 8 – 6 – 1 – 5 – 8 – 6

3. **In your own words, what is the seed for this random number generator? You must provide the correct values for both pieces used in the November 11th pull and explain the calculation made to earn full points. Why is this a good or bad thing? Justify your answer.** (15 total points)
   The random number generator is based on a predictable seed which include the date and time represented as seconds in code. Since we know the lottery number from November 11, we just need to identify the time from the 86400 potential seeds, combined with the date will generate the sequence. Once we get the correct timeInSeconds, we reuse the time to predict the November 12 winning ticket.
   This is a bad design for a lottery since a lottery should be completely random with no predictability. This lottery is based on the timeInSeconds from lottery pick from day before which are values that can be predictable. This lets an attacker be able to brute force and get future winning numbers.

4. **What would be an alternate way to seed this random number generator such that it couldn't be predicted either forward or backward? Justify your answer.** (10 total points)
   An alternate way to seed this random number generator so it cant be predicted in any event is having a true random data source, as well as applying cryptographic hashing. Using data from a quantum random number generator source is unpredictable with a random process to reduce the predictability. Applying cryptographic hash function like SHA512 to input times and other keys or data before using as seed will reduce the predictability of it.